4/5/1 **(Item 1 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

015080421 **Image available**
WPI Acc No: 2003-140939/200313
XRPX Acc No: N03-111882
Debugging **a platform-independent** virtual machine **within a computer systems, where an agent on the platform-independent** virtual machine **provides a set of functions for accessing variables in the platform**
Patent Assignee: SUN MICROSYSTEMS INC (SUNM); SOKOLOV S (SOKO-I); WALLMAN D (WALL-I)
Inventor: SOKOLOV S; WALLMAN D
Number of Countries: 100 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200303215	A2	20030109	WO 2002US19690	A	20020621	200313 B
US 20030028861	A1	20030206	US 2001895903	A	20010628	200313
AU 2002316324	A1	20030303	AU 2002316324	A	20020621	200452

Priority Applications (No Type Date): US 2001895903 A 20010628
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200303215	A2	E	16	G06F-011/36	

Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO RU SD SE SG SI SK SL TJ TM TN TR TT TZ UA UG UZ VN YU ZA ZM ZW
Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZM ZW

US 20030028861	A1		G06F-009/44	
AU 2002316324	A1		G06F-011/36	Based on patent WO 200303215

Abstract (Basic): WO 2003003215 A2
NOVELTY - An agent is employed on the platform-independent **virtual machine** , which provides a set of functions for accessing variables in the platform-independent **virtual machine** . The agent examines the **current state** of the variables in the platform-independent **virtual machine** and informs the host machine. An operator of the host machine can then analyze the **current state** of the variables.
DETAILED DESCRIPTION - INDEPENDENT CLAIM included for the following:computer-readable storage medium; apparatus
USE - For computer systems, PDAs etc.
ADVANTAGE - Facilitates **debugging** a platform-independent **virtual machine** .
DESCRIPTION OF DRAWING(S) - The diagram shows target machine
target (108)
platform-independent **virtual machine** (202)
agent (204)
pp; 16 DwgNo 2/4
Title Terms: **DEBUG** ; PLATFORM; INDEPENDENT; VIRTUAL; MACHINE; COMPUTER; SYSTEM; AGENT; PLATFORM; INDEPENDENT; VIRTUAL; MACHINE; SET; FUNCTION; ACCESS; VARIABLE; PLATFORM
Derwent Class: T01
International Patent Class (Main): G06F-009/44; G06F-011/36
File Segment: EPI

03250844 **Image available**

COMPUTER SYSTEM

PUB. NO.: 02-226344 [JP 2226344 A]
PUBLISHED: September 07, 1990 (19900907)
INVENTOR(s): MIYAMOTO SHINICHI
SHIMIZU TAKESHI
APPLICANT(s): FUJI XEROX CO LTD [359761] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 01-047976 [JP 8947976]
FILED: February 27, 1989 (19890227)
INTL CLASS: [5] G06F-011/14; G06F-009/46; G06F-011/28
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1135, Vol. 14, No. 534, Pg. 45, November 26, 1990 (19901126)

ABSTRACT

PURPOSE: To improve the efficiency of **debugging** processing by referring to and changing states of a virtual memory and a **virtual machine** in accordance with an inputted command at the time of the occurrence of error, by which execution of a program cannot be continued, in the **virtual machine** .

CONSTITUTION: A **virtual machine** part 2 reads out the program from a virtual storage part 1 and interprets and executes it, and the **virtual machine** part 2 outputs error information when recognizing the occurrence of error, by which execution of the program cannot be continued, in its own **virtual machine** during the execution of the program. A **debugger** command executing part 5 of a **debugger** 3 refers to and changes values of **variables** , states of stacks, etc., of the virtual storage part 1 in accordance with the command. When error information is inputted from the **virtual machine** part 2 to an error detecting part 6, this part 6 indicates switching to an input/output switching part 7, and this part 7 disconnects the **virtual machine** part 2 and an input/output part 8 and connects the **debugger** 3 and the input/output part 8. Thus, the efficiency of **debugging** processing is improved.

10/5/4 **(Item 3 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

014009405 **Image available**
WPI Acc No: 2001-493619/200154
XRPX Acc No: N01-365506

Debug **system of routine-work program for handheld terminals, performs symbolic** debug **by integrating** debug **function in interpreter and communicating with host computer and handheld terminal**

Patent Assignee: PENTEL KK (PENL)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 2001184230	A	20010706	JP 99370382	A	19991227	200154	B

Priority Applications (No Type Date): JP 99370382 A 19991227
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 2001184230	A		12	G06F-011/28	

Abstract (Basic): JP 2001184230 A

NOVELTY - An interpreter performs interpretation execution of a **virtual machine** language in a handheld terminal (2) using a compiler which outputs the **virtual machine** language. A symbolic **debug** is performed by integrating a **debug** function in an interpreter and communicating with a host computer and the handheld terminal.

USE - **Debug** system of routine-work program for handheld terminals.

ADVANTAGE - **Debug** function is performed without applying load to the interpreter and the execution program, beside handheld terminal. The operation of handheld terminal and the display of **variable** are confirmed by the co-operation with a host computer and source program, without performing re-compiler to symbolic **debugs** , thereby simple **debug** operation is obtained.

DESCRIPTION OF DRAWING(S) - The figure shows the perspective diagram of handheld terminal. (Drawing includes non-English language text).

Handheld terminal (2)

pp; 12 DwgNo 2/17

Title Terms: **DEBUG** ; SYSTEM; ROUTINE; WORK; PROGRAM; TERMINAL; PERFORMANCE ; SYMBOL; **DEBUG** ; INTEGRATE; **DEBUG** ; FUNCTION; INTERPRETATION; COMMUNICATE; HOST; COMPUTER; TERMINAL
Derwent Class: T01
International Patent Class (Main): G06F-011/28
International Patent Class (Additional): G06F-009/445
File Segment: EPI

```
Set      Items   Description
S1       11313   DEBUG? OR (BUG OR FAULT OR FLAW OR ERROR)(N)(DETECT? OR FI-
             ND? OR LOCAT? OR CORRECT?)(7N)(SOFTWARE? OR TOOL? OR MODULE? -
             OR APPLICATION?)
S2        1322   VIRTUAL()MACHINE?
S3       14735   (VARIABLE OR CURRENT)(2N)(STATE? OR STATUS)
S4           1   S1 AND S2 AND S3
S5       62299   SOFTWARE()(ROBOT? OR AGENT?) OR IA OR INTELLIGENT()AGENT? -
             OR SOFTBOT?
S6          53   S3 AND S5
S7           0   S6 AND (VM OR VIRTUAL()MACHINE? OR HANDHELD? OR PORTABLE? -
              OR MOBILE? OR CELLPHONE? OR PDA OR PERSONAL()DIGITAL()ASSISTA-
              NT? OR PALM OR PALMTOP )
S8           2   S6 AND IC=(G06F-009? OR G06F-011?)
S9           5   S1 AND S2 AND VARIABL?
S10          4   S9 NOT (S4 OR S8)
S11     142732   DEBUG? OR (BUG OR BUGS OR FAULT? OR FLAW? OR ERROR?)(2N)(D-
             ETECT? OR IDENTIF? OR FIND? OR LOCAT? OR CORRECT?) OR DEBUG? -
             OR EMULAT?
S12        312   S11 AND S3
S13          0   S12 AND S5
S14         11   S12 AND (HANDHELD? OR PALMTOP? OR PALM OR PDA OR CELLPHONE?
               OR SMARTPHONE? OR MOBILE? OR PDA OR PERSONAL()DIGITAL()ASSIS-
              TANT? OR PALM)
S15          0   S11 AND S3 AND S5
S16        312   S11 AND S3
S17          2   (VM OR S2 OR JAVA OR ACTIVE()X OR APPLET?) AND S16
S18         13   S14 OR S17
S19         12   S18 NOT (S8 OR S9 OR S10)
S20      40923   S11 AND (PROGRAM? OR CODE? OR SOFTWARE? OR APPLICATION? OR
             OS OR OPERATING()SYSTEM?)
S21        133   S20 AND (STATE OR STATUS)(3N)(CURRENT? OR VARIABLE?)
S22          1   S21 AND (VM OR S2 OR JAVA OR ACTIVE()X OR APPLET?)
S23          0   S22 NOT S19
File 347:JAPIO Nov 1976-2004/Oct(Updated 050208)
         (c) 2005 JPO & JAPIO
File 350:Derwent WPIX 1963-2005/UD,UM &UP=200510
         (c) 2005  Thomson Derwent
```

23/5/1 **(Item 1 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

016755695 **Image available**
WPI Acc No: 2005-079973/200509
XRPX Acc No: N05-070338

Computer e.g. desktop computer, program, error e.g. syntax error , identifying method, involves responding to receiving step-back command from user input device by recalling and displaying instruction that generated exception

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: MORGAN F F
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20040268310	A1	20041230	US 2003610187	A	20030630	200509	B

Priority Applications (No Type Date): US 2003610187 A 20030630
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20040268310	A1		7	G06F-009/44	

Abstract (Basic): US 20040268310 A1

NOVELTY - The method involves responding to receiving a step-back command from a user input device by recalling and displaying an error-checking instruction that generated an exception. A **current** program **state** of a target computer program is reset to match a program state of the target program at which the exception was generated. The error-checking instruction is checked to determine if the exception has been generated.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(A) a programmable apparatus for **identifying errors** in a computer program

(B) a computer readable memory for causing a computer to **identify errors** in a target program having an error- checking instruction.

USE - Used for **identifying** an **error** e.g. syntax error and logical error, in a program of a computer such as desktop computer, notebook computer, **personal digital assistants** (**PDA**), server, and **handheld** computer.

ADVANTAGE - The step-back command allows a programmer to readily step backwards through instructions in the computer program to **identify errors** in the computer program without added cost and overhead using additional files. The method thus allows a programmer to isolate the cause of the exception, without the undue repetition.

DESCRIPTION OF DRAWING(S) - The drawing shows a flowchart of a method of **identifying** an **error** in a target computer program.

pp; 7 DwgNo 3/3

Title Terms: COMPUTER; COMPUTER; PROGRAM; ERROR; SYNTAX; ERROR; IDENTIFY; METHOD; RESPOND; RECEIVE; STEP; BACK; COMMAND; USER; INPUT; DEVICE; RECALL; DISPLAY; INSTRUCTION; GENERATE
Derwent Class: T01
International Patent Class (Main): G06F-009/44
File Segment: EPI

23/5/4 **(Item 4 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

014564620 **Image available**
WPI Acc No: 2002-385323/200242
XRPX Acc No: N02-301718
State detection method for hierarchical networks, involves detecting states to be allocated to individual network elements based on status message received from respective elements
Patent Assignee: LUCENT TECHNOLOGIES INC (LUCE)
Inventor: KETTSCHAU H J; REICHENBACH J A
Number of Countries: 025 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 1191803	A1	20020327	EP 2000308190	A	20000920	200242	B

Priority Applications (No Type Date): EP 2000308190 A 20000920
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 1191803	A1	E	13	H04Q-007/34	

Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI

Abstract (Basic): EP 1191803 A1
NOVELTY - The method involves detecting the states which can be allocated to unambiguously addressed individual network elements, based on the status message received from the respective elements. The detected states are signaled depending on the network topology.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:
(a) Network-state detecting system; Computer program for network state detection
USE - For detecting state of hierarchially-structured network such as **mobile** radio network.
ADVANTAGE - Enables **detecting fault** messages of the individual network elements more efficiently. Signaling based on network topology reduces indication of redundant information and ensures direct detection of **current** network **state** .
DESCRIPTION OF DRAWING(S) - The figure shows a section of the hierarchially structured arrangement of network elements
pp; 13 DwgNo 2/5
Title Terms: STATE; DETECT; METHOD; HIERARCHY; NETWORK; DETECT; STATE; ALLOCATE; INDIVIDUAL; NETWORK; ELEMENT; BASED; STATUS; MESSAGE; RECEIVE; RESPECTIVE; ELEMENT
Derwent Class: T01; W01
International Patent Class (Main): H04Q-007/34
International Patent Class (Additional): H04M-003/22
File Segment: EPI

23/5/6 **(Item 6 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

013703446 **Image available**
WPI Acc No: 2001-187670/200119
XRPX Acc No: N01-134539

Error detector in mobile communication network, has controller which controls light emission condition of light emitting element corresponding to count value output by counter

Patent Assignee: ANRITSU CORP (ANRI)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 2001016187	A	20010119	JP 99185139	A	19990630	200119	B

Priority Applications (No Type Date): JP 99185139 A 19990630
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 2001016187	A		6	H04L-001/00	

Abstract (Basic): JP 2001016187 A

NOVELTY - Signal of fixed period is counted n' times and count value (CN) is output. Circulation counter (21) repeats counting for preset period T0', based on which light emitting element (13) maintains **variable** light emission **state** . The counter is set at start of bit error measurement and stopped after measurement. A controller controls light emission of element (13) corresponding to output of count value.

USE - For **mobile** communication network, internet, public telephone network, satellite circuit network, etc.

ADVANTAGE - As area of console panel is reduced, size of entire apparatus becomes small. Damaging the function of warning can be prevented due to repeating change of light emission condition of light emitting element throughout.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram showing schematic component of **error detector** .

Light emitting element (13)
Circulation counter (21)
Count value (CN)
pp; 6 DwgNo 2/7

Title Terms: ERROR; DETECT; **MOBILE** ; COMMUNICATE; NETWORK; CONTROL; CONTROL; LIGHT; EMIT; CONDITION; LIGHT; EMIT; ELEMENT; CORRESPOND; COUNT; VALUE; OUTPUT; COUNTER
Derwent Class: W01; W02; W05
International Patent Class (Main): H04L-001/00
International Patent Class (Additional): G01D-021/00; G08C-025/00; H04L-025/02; H04L-029/14
File Segment: EPI

```
Set     Items   Description
S1      11313   DEBUG? OR (BUG OR FAULT OR FLAW OR ERROR)(N)(DETECT? OR FI-
             ND? OR LOCAT? OR CORRECT?)(7N)(SOFTWARE? OR TOOL? OR MODULE? -
             OR APPLICATION?)
S2       1322   VIRTUAL()MACHINE?
S3      14735   (VARIABLE OR CURRENT)(2N)(STATE? OR STATUS)
S4          1   S1 AND S2 AND S3
S5      62299   SOFTWARE()(ROBOT? OR AGENT?) OR IA OR INTELLIGENT()AGENT? -
             OR SOFTBOT?
S6         53   S3 AND S5
S7          0   S6 AND (VM OR VIRTUAL()MACHINE? OR HANDHELD? OR PORTABLE? -
             OR MOBILE? OR CELLPHONE? OR PDA OR PERSONAL()DIGITAL()ASSISTA-
             NT? OR PALM OR PALMTOP )
S8          2   S6 AND IC=(G06F-009? OR G06F-011?)
S9          5   S1 AND S2 AND VARIABL?
S10         4   S9 NOT (S4 OR S8)
S11    142732   DEBUG? OR (BUG OR BUGS OR FAULT? OR FLAW? OR ERROR?)(2N)(D-
             ETECT? OR IDENTIF? OR FIND? OR LOCAT? OR CORRECT?) OR DEBUG? -
             OR EMULAT?
S12       312   S11 AND S3
S13         0   S12 AND S5
S14        11   S12 AND (HANDHELD? OR PALMTOP? OR PALM OR PDA OR CELLPHONE?
              OR SMARTPHONE? OR MOBILE? OR PDA OR PERSONAL()DIGITAL()ASSIS-
             TANT? OR PALM)
S15         0   S11 AND S3 AND S5
S16       312   S11 AND S3
S17         2   (VM OR S2 OR JAVA OR ACTIVE()X OR APPLET?) AND S16
S18        13   S14 OR S17
S19        12   S18 NOT (S8 OR S9 OR S10)
S20     40923   S11 AND (PROGRAM? OR CODE? OR SOFTWARE? OR APPLICATION? OR
             OS OR OPERATING()SYSTEM?)
S21       133   S20 AND (STATE OR STATUS)(3N)(CURRENT? OR VARIABLE?)
S22         1   S21 AND (VM OR S2 OR JAVA OR ACTIVE()X OR APPLET?)
S23         0   S22 NOT S19
S24        93   S16 AND IC=(G06F-009? OR G06F-011?)
S25         0   S24 AND (LIMITED OR LEAST OR SMALL OR MINOR OR LESS)(2N)(M-
             EMOR? OR STORAGE? OR RAM OR ROM)
S26        48   S24 AND (RESOURC? OR MEMOR? OR STORAGE? OR CACHE? OR BUFFE-
             R? OR RAM OR ROM OR PROM OR EPROM OR EEPROM)
S27        45   S26 NOT AD=20010628:20030628
S28        44   S27 NOT AD=20030628:20050301
S29        43   S28 NOT (S8 OR S19 OR S9 OR S10)
S30         5   S29 AND IC=(G06F-009/44 OR G06F-011/36)
S31    103858   MC=(T01-F05G3 OR T01-J20C OR T01-S03)
S32         7   S31 AND S29
S33        12   S30 OR S32
File 347:JAPIO Nov 1976-2004/Oct(Updated 050208)
         (c) 2005 JPO & JAPIO
File 350:Derwent WPIX 1963-2005/UD,UM &UP=200510
         (c) 2005  Thomson Derwent
```

33/5/2 **(Item 2 from file: 347)**
DIALOG(R)File 347:JAPIO

01988141 **Image available**
DEBUG SYSTEM FOR HIGH-LEVEL LANGUAGE PROGRAM

PUB. NO.: 61-202241 [JP 61202241 A]
PUBLISHED: September 08, 1986 (19860908)
INVENTOR(s): KINUGASA TAKASHI
KOBAYASHI KENZO
KAMIHATSU KAZUMI
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 60-044095 [JP 8544095]
FILED: March 06, 1985 (19850306)
INTL CLASS: [4] **G06F-011/28** ; **G06F-009/44**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 541, Vol. 11, No. 32, Pg. 103, January 30, 1987 (19870130)

ABSTRACT

PURPOSE: To monitor a specific variable and to trace the state shift of the variable by storing the states of variables in a variable control area according to the tracing conditions shown by a control flag.

CONSTITUTION: A BASIC interpreter 9 extracts a command processing routine corresponding to a command out of a command processing subroutine group 4 and executes it through a decoding/executing part 2. Here a variable control flag 8 on a BASIC interpreter **memory** map (b) is turned on, the interpreter 9 is set under a **variable** control **state** . Then a **variable** is retrieved by a variable control processing part 5. If the corresponding variable is registered, a subrou tine is executed according to the tracing conditions shown by a control flag 8'. Then the **state** of the **variable** is stored in a variable control area 7. Thus the executing **state** of the **variable** designated by the flag 8' can be known by referring to the contents of the area 7 after execution of the subroutine.

33/5/5 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX

013571098 **Image available**
WPI Acc No: 2001-055305/200107
XRPX Acc No: N01-042810

Source code debugger for C-language, has coverage status monitor detecting whether the updated value of the variable is saved on the access status and displays the new and old variable value

Patent Assignee: NEC CORP (NIDE)
Number of Countries: 001 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 2000315166	A	20001114	JP 99123880	A	19990430	200107	B
JP 3298554	B2	20020702	JP 99123880	A	19990430	200246	

Priority Applications (No Type Date): JP 99123880 A 19990430
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 2000315166	A		11	G06F-011/28	
JP 3298554	B2		11	G06F-011/28	Previous Publ. patent JP 2000315166

Abstract (Basic): JP 2000315166 A

NOVELTY - An **emulation memory** (101) stores the **debugged** variable and a coverage **memory** (102) stores the access **status** to the **variable** in **emulation memory** during the program execution. A coverage status monitor (110) monitors the access status and detects whether the updated value of the variable is saved on the access data. The new and old value of the **debugged** variable is displayed.

USE - For detecting modification of strange numerical value in the source program of C-language.

ADVANTAGE - Modification of strange numerical value to the name value can also be detected, since write-in status of **storage memory** is monitored. The change value of the variable is displayed along with old value, since the coverage **memory** specifies variable for which value is changed.

DESCRIPTION OF DRAWING(S) - The figure shows block diagram showing the component of the source data **debugger** .

Emulation memory (101)
Coverage **memory** (102)
Status monitor (110)
pp; 11 DwgNo 1/8

Title Terms: SOURCE; CODE; LANGUAGE; COVER; STATUS; MONITOR; DETECT; UPDATE ; VALUE; VARIABLE; SAVE; ACCESS; STATUS; DISPLAY; NEW; VARIABLE; VALUE
Derwent Class: T01
International Patent Class (Main): **G06F-011/28**
File Segment: EPI

33/5/6 **(Item 4 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

012263923 **Image available**
WPI Acc No: 1999-070029/199906
XRPX Acc No: N99-051323

Memory **monitoring and** debugging **module for POST and BIOS code in PC - responds to commands from external for monitoring locations of** memory **and execute** PROM **code instructions**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: CRUMP D T; PANCOAST S T

Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5850562	A	19981215	US 94266927	A	19940627	199906	B

Priority Applications (No Type Date): US 94266927 A 19940627
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5850562	A		18	G06F-001/24	

Abstract (Basic): US 5850562 A

The module is stored in a **PROM** and is executed before the OS is loaded into **RAM** and executed. Monitor commands are received from external responsive to which signal representing **current state** of CPU registers is transmitted to the external commanding unit. The state of the registers are modified. The **current state** of a **memory** location is notified to the commanding unit in response to its command, and the state of the location is modified on receiving further command.

Likewise, **debugger** commands received from the external are also responded to, by setting one breakpoint corresponding to a specific location in **RAM** and clearing it. The execution of **PROM** code by the CPU, at a specific code instruction is also triggered by an external command. The execution of the code allows access to portion of **RAM** corresponding to breakpoint, after the execution of the **PROM** code is stopped by the CPU. The CPU is also initiated to execute single **PROM** code instruction and transmit **current state** of register of CPU to external commanding unit.

ADVANTAGE - Provides low level monitor and **debugger** which is executed before completion of POST code execution. Enhances productivity of POST and BIOS code. Is easily invoked without need for keyboard, video subsystem and disk drive to be functional.

Dwg.4/5

Title Terms: **MEMORY** ; MONITOR; **DEBUG** ; MODULE; POST; CODE; RESPOND; COMMAND; EXTERNAL; MONITOR; LOCATE; **MEMORY** ; EXECUTE; **PROM** ; CODE; INSTRUCTION
Derwent Class: T01
International Patent Class (Main): G06F-001/24
International Patent Class (Additional): **G06F-009/24** ; **G06F-011/28** ; **G06F-011/34**
File Segment: EPI

```
Set     Items   Description
S1      96162   VIRTUAL()MACHINE? OR VM? ? OR JAVA OR ACTIVE()X OR APPLET?
             OR JVM
S2     157287   (VARIABLE? OR CURRENT)(2N)(STATE? OR STATUS)
S3      72543   SOFTWARE()(ROBOT? OR AGENT?) OR IA OR INTELLIGENT()AGENT? -
             OR SOFTBOT? OR BOT OR BOTS
S4     640674   HANDHELD? OR PORTABLE? OR MOBILE? OR CELLPHONE? OR PDA OR -
             PERSONAL()DIGITAL()ASSISTANT? OR PALM OR PALMTOP
S5     288162   DEBUG? OR (BUG OR BUGS OR FAULT? OR FLAW? OR ERROR?)(2N)(D-
             ETECT? OR IDENTIF? OR FIND? OR LOCAT? OR CORRECT?) OR DEBUG? -
             OR EMULAT?
S6       8202   S5 AND (HANDHELD? OR PALMTOP? OR PALM OR PDA OR CELLPHONE?
             OR SMARTPHONE? OR MOBILE? OR PDA OR PERSONAL()DIGITAL()ASSIST-
             ANT? OR PALM)
S7       1835   (LOW()LEVEL OR SIMPLE OR PRIMITIVE?)(3N)S5
S8         16   S2 AND S7
S9       1478   S2 AND S5
S10        19   S9 AND S1
S11         3   S9 AND S3
S12        42   S9 AND S4
S13        50   S7 AND S6
S14       125   S8 OR S10 OR S11 OR S12 OR S13
S15        84   RD (unique items)
S16        66   S15 NOT PY>2001
S17        41   S16 AND (VARIABLE? OR STATE? OR STATUS)
S18        41   S17 NOT PD=20010628:20040628
S19        41   S18 NOT PD=20040628:20050301
File   8:Ei Compendex(R) 1970-2005/Jan W3
         (c) 2005 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2005/Jan
         (c) 2005 ProQuest Info&Learning
File  65:Inside Conferences 1993-2005/Feb W2
         (c) 2005 BLDSC all rts. reserv.
File   2:INSPEC 1969-2005/Feb W1
         (c) 2005 Institution of Electrical Engineers
File  94:JICST-EPlus 1985-2005/Jan W1
         (c)2005 Japan Science and Tech Corp(JST)
File 111:TGG Natl.Newspaper Index(SM) 1979-2005/Feb 11
         (c) 2005 The Gale Group
File   6:NTIS 1964-2005/Feb W1
         (c) 2005 NTIS, Intl Cpyrght All Rights Res
File 144:Pascal 1973-2005/Feb W1
         (c) 2005 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File  34:SciSearch(R) Cited Ref Sci 1990-2005/Feb W2
         (c) 2005 Inst for Sci Info
File  99:Wilson Appl. Sci & Tech Abs 1983-2005/Jan
         (c) 2005 The HW Wilson Co.
File  95:TEME-Technology & Management 1989-2005/Jan W2
         (c) 2005 FIZ TECHNIK
```

19/5/34 (Item 1 from file: 6)
DIALOG(R)File 6:NTIS

1749289 NTIS Accession Number: AD-A266 689/9
Xab: A Tool for Monitoring PVM Programs
Beguelin, A. L.
Carnegie-Mellon Univ., Pittsburgh, PA. School of Computer Science.
Corp. Source Codes: 005343049; 423887
Report No.: CMU-CS-93-164
2 Jun 93 10p
Languages: English
Journal Announcement: GRAI9321
Order this product from NTIS by: phone at 1-800-553-NTIS (U.S. customers); (703)605-6000 (other countries); fax at (703)321-8547; and email at orders@ntis.fedworld.gov. NTIS is located at 5285 Port Royal Road, Springfield, VA, 22161, USA.
NTIS Prices: PC A02/MF A01
Country of Publication: United States
Xab (X-window Analysis and **deBugging**) is a tool for run time monitoring of PVM (Parallel **Virtual Machine**) programs. PVM supports the programming of a network of heterogeneous computers as a single parallel computer. Using Xab, PVM programs can easily be instrumented and monitored. Xab uses PVM to monitor PVM programs. This makes Xab very **portable** but it leads to interesting issues of how to make Xab peacefully coincide with the programs it monitors. Xab consists of three main components, a user library, a monitoring program, and an X windows front end. The user library provides instrumented versions of the PVM calls. The monitoring program runs as a PVM process and gathers monitor events in the form of PVM messages. The Xab front end displays information graphically about PVM processes and messages. This paper discusses the design, implementation. and use of the Xab tool. Related work is briefly presented and contrasted with the approach taken with Xab. How Xab works and how it is used are discussed in detail. Finally, the **current status** of Xab is presented along with future directions of where the research may go from here....Monitoring, Parallel programming, **Debugging** , Real time.
Descriptors: *Monitoring; *Computer program verification; Computer programming; Computers; Libraries; Networks; Real time; Windows; Computer networks
Identifiers: *Software tools; PVM(Parallel **Virtual Machine**); NTISDODXA
Section Headings: 62B (Computers, Control, and Information Theory--Computer Software)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1798	717/124-135.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 10:33
L2	1	1 and (restrict$3 with output with state)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 10:43
L3	86	1 and ((reduced OR limited) near2 memory)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 10:57
L4	21	3 and ((variable OR object) with state)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 11:15
L5	7	(low with memory with debugger)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 11:23
L6	6	1 and (low with memory with over$1head)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 11:23
L7	220	1 and (hand$1held OR pda or palm)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 11:23
L8	74	7 and ((virtual adj machine) or vm)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 11:24
L9	7	8 and ((object OR variable) near2 state)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 13:16
L10	9	1 and (record$3 with state with variables)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 13:18

L11	60	((handheld OR mobile OR pda OR palm) same (debug$4 OR record$3) same (information OR data)) and agent and (state with variable)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 13:20
L12	13	11 and ((virtual adj machine) OR $1vm)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 13:20
L13	2	1 and (agent same current same variable same state)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 13:22
L14	4	1 and (agent same variable same state)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 14:55
L15	0	(simple with debug$4 with operationg)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 15:05
L16	18	(handheld with debug$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 15:11
L17	11	16 and state	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 15:13
L18	2	"6795962".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 15:13
L19	7	("5093914" | "5317740" | "5555419" | "5659753" | "5781778" | "5848274" | "6151701").PN. OR ("6795962").URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2005/02/16 15:25
S1	28	"0129337"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/07 12:06

S2	3	717/124.ccls. and evans.in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/07 14:14
S3	45	(virtual same machine same debug$4).ab.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/20 11:29
S4	9	(virtual same machine same debug$4).ti.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/20 11:30
S5	24	((virtual adj machine) with debug$4).ab.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/20 11:30
S6	1798	717/124-135.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/16 10:32
S7	55	S6 and (current with state with (objects OR variables))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/15 17:12
S8	2	S6 and (current with state with (objects OR variables) with limit$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/15 16:58
S9	283	S6 and (limit$3 with memory)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/15 17:12
S10	64	S9 and (variable with state)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/15 17:17
S11	26	S10 and (vm OR (virtual adj machine))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/15 17:17